

The Performances Study of EDCF with Block_Ack in WLANs

Chien-Erh Weng, Chien-Hung Chen, Chiung-Hsing Chen, Jyh-Horng Wen

► **To cite this version:**

Chien-Erh Weng, Chien-Hung Chen, Chiung-Hsing Chen, Jyh-Horng Wen. The Performances Study of EDCF with Block_Ack in WLANs. James J. Park; Albert Zomaya; Sang-Soo Yeo; Sartaj Sahni. 9th International Conference on Network and Parallel Computing (NPC), Sep 2012, Gwangju, South Korea. Springer, Lecture Notes in Computer Science, LNCS-7513, pp.328-335, 2012, Network and Parallel Computing. <10.1007/978-3-642-35606-3_39>. <hal-01551375>

HAL Id: hal-01551375

<https://hal.inria.fr/hal-01551375>

Submitted on 30 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



The Performances Study of EDCF with Block_Ack in WLANs

Chien-Erh Weng^{1*}, Chien-Hung Chen¹, Chiung-Hsing Chen¹, and Jyh-Horng Wen²

¹Department of Electronic Communication Engineering,
National Kaohsiung Marine University
Kaohsiung, Taiwan. ROC
ceweng@mail.nkmu.edu.tw

* Correspondence addressee

²Department of Electrical Engineering,
Tunghai University
Taichung, Taiwan. ROC
jhwen@thu.edu.tw

Abstract. IEEE 802.11e Enhanced Distributed Coordination Function (EDCF) is a new wireless technology in wireless local area networks (WLANs). It defines a new supplement to the existing IEEE 802.11 MAC protocol. In IEEE 802.11e EDCF, the aim is providing a QoS support in WLANs. While the system services different Access categories (ACs), IEEE 802.11e EDCF does not perform well under high load conditions. In order to improve the efficiency, we pay attention to the EDCF of IEEE 802.11e with Block_Ack mechanism. We first proposed a markov chain model and studied the behavior. We extend the model to support IEEE 802.11e EDCF, and presented a more accurate analysis under non-ideal channel environment. We also compared it with that without Block_Ack mechanism under channel error environment.

Keywords. EDCF, Acs, QoS, Block_Ack

1 INTRODUCTION

In recent years, the WLANs market is experiencing an explosive growth. The medium access control (MAC) protocol is the key element that provides the efficiency in accessing the channel, while satisfying the quality of service (QoS) requirements. IEEE 802.11e EDCF is a new wireless technology which is an enhanced version of IEEE 802.11 distributed coordination function (DCF). The IEEE 802.11e EDCF aims at improving the capabilities and efficiency of the IEEE 802.11 MAC protocol by defining a new mechanism to support the QoS services. While the system services different ACs, EDCF does not perform well under high load conditions. In order to improve the efficiency, EDCF provides two mechanisms named as transmission opportunity (TXOP) and Block_ACK. These two mechanisms are allowed to offer new data

transmission services that include the multiple frame delivery [1-3]. There have been many performance analyses of the IEEE 802.11e EDCF. Deng and Chang [4] proposed a priority scheme by differentiating the backoff window. Aad and Castelluccia [5] proposed a priority scheme by differentiating inter-frame spaces (IFS's), in which a higher priority class uses IFS, whereas a lower priority class uses a space that equals the sum of IFS and the maximum window size. In [6] Veres and Campbell et al. proposed priority schemes by differentiating the minimum backoff window size and the maximum window size. These performance studies neglected that high load situation. They can't truly reflect the real operation of EDCF with priority schemes. In order to improve the efficiency, IEEE 802.11e EDCF provides two mechanisms named as TXOP and Block_Ack mechanisms. In this paper, we pay attention to the EDCF of IEEE 802.11e with Block_Ack mechanism. We proposed a markov chain model and studied the behavior. We extend the model to support IEEE 802.11e EDCF, and present a more accurate analysis of the IEEE 802.11e EDCF under a non-ideal channel environment. We also compared it with that without Block_Ack mechanism under channel error environment.

The rest of this paper is organized as follows. In section 2, a general description of our proposed model with Block_Ack mechanism is presented. Analytical performance deviations of proposed model including throughput analysis under non-ideal channel scenario are presented in section 3. The numerical results are given with discussion in section 4. Finally, conclusions are drawn in section 5.

2 Enhanced DCF of IEEE 802.11e

IEEE 802.11e EDCF is a new wireless technology which is an enhanced version of IEEE 802.11 DCF. It defines a new supplement to the existing IEEE 802.11 MAC protocol. The IEEE 802.11e EDCF aims at improving the capabilities and efficiency of the IEEE 802.11 MAC protocol by defining a new mechanism to support the QoS services.

2.1 IEEE 802.11e EDCF Scheme

EDCF specifies four default access categories (ACs). Each STA contends for the channel access and independently starts its backoff depending on its associated AC. Each AC uses AIFS[AC], $CW_{min}[AC]$ and $CW_{max}[AC]$ instead of DIFS, CW_{min} and CW_{max} of the DCF [11].

The contention method of EDCF is the same as that in DCF. Each STA having a frame to transmit has to wait for the channel to be idle without interruption for a period AIFS[AC], and then it should start a random backoff process with its own $CW[AC]$. For each time slot interval, during which the channel stays idle, the random backoff value is decremented. When the backoff counter reaches zero, the frame is transmitted. AIFS[AC] is calculated as follows:

$$AIFS[AC] = AIFSN[AC] \times aSlotTime + aSIFSTime, \quad (1)$$

where the backoff time is calculated as follows:

$$\text{backoff time} = \text{random_integer} \times \text{aSlotTime}, \quad (2)$$

where random_integer is uniformly and randomly chosen in the range $[0, CW(AC)]$, instead of $[0, W-1]$ in the DCF. The value of CW_{min} is 15 and CW_{max} is 1023 [12]. Initially, CW of each AC is equal to $CW_{min}[AC]$. After each collision, CW is doubled up to:

$$CW_{max}[AC] = 2^m \times (CW_{min}[AC]), \quad (3)$$

where m is called the maximum backoff stage. Once it reaches $CW_{max}[AC]$, it remains at this value until it is reset.

2.2 The System Model

We assume in the following that for a given station in the priority i class, $b(i, t)$ is defined as a random process representing the value of backoff counter at time t , and $s(i, t)$ is defined as the random process representing the backoff stage j , where $0 \leq j \leq m$ and m is the maximum backoff stage. The value of the backoff counter is uniformly chosen in the range $(0, 1, \dots, W_{i,j} - 1)$, where $W_{i,j} = 2^j W_{i,0}$ and $W_{i,0} = CW_{min}[i]$.

Let p_i denote the probability that a transmitted frame collides in the priority i class. A Markov chain $\{s(i, t), b(i, t)\}$ can be established to analyze the contention process. Therefore, the state of each station in the priority i class is described by $\{i, j, k\}$, where i is just an index standing for the priority i class, j stands for the backoff stage and takes values $(0, 1, \dots, m)$, and k stands for the backoff delay and takes values $(0, 1, \dots, W_{i,j} - 1)$ in time slots. The state transition diagram for the priority i class is shown in Fig. 1, where the state $\{i, -1, 0\}$ stands for the state that the station senses when the channel is idle and when a previous frame transmits successfully, the STA can transmit a frame immediately without activating the backoff stage. The transmission probabilities are listed as follows:

1. The backoff counter freezes when the STA senses that the channel is busy in the priority i class:

$$P\{i, j, k | i, j, k\} = p_i, \quad 0 \leq k \leq W_{i,j} - 1, \quad 0 \leq j \leq m. \quad (4)$$

2. The backoff counter decrements when the STA senses the channel is idle in the priority i class:

$$P\{i, j, k | i, j, k + 1\} = 1 - p_i, \quad 0 \leq k \leq W_{i,j} - 2, \quad 0 \leq j \leq m. \quad (5)$$

3. The STA enters the $\{i, -1, 0\}$ state if it has a successful transmission with the previous frame in the priority i class:

$$P\{i, -1, 0 | i, j, 0\} = 1 - p_i, \quad 0 \leq j \leq m. \quad (6)$$

4. The STA transmits its frame without entering the backoff process at $\{i, -1, 0\}$ state if it detects that its previous transmitted frame was successfully received and the channel is idle in the priority i class:

$$P\{i, -1, 0 | i, -1, 0\} = 1 - p_i. \quad (7)$$

5. The STA defers the transmission of a new frame and enters stage 0 of the backoff process if the STA finds a collision has occurred at $\{i, -1, 0\}$ state in the priority i class:

$$P\{i, 0, k | i, -1, 0\} = p_i / W_{i,0}, \quad 0 \leq k \leq W_{i,0} - 1. \quad (8)$$

6. The STA chooses a backoff delay of the next stage j after an unsuccessful transmission at stage $j-1$ in the priority i class:

$$P\{i, j, k | i, j-1, 0\} = p_i / W_{i,j}, \quad 1 \leq j \leq m, \quad 0 \leq k \leq W_{i,j} - 1. \quad (9)$$

Let $b_{i,j,k} = \lim P_r\{s(i,t) = j, b(i,t) = k\}$ be the stationary distribution of the Markov chain. We can derive the following relations by chain regularities:

$$b_{i,-1,0} = \frac{2p_i(1-p_i)^2(1-2p_i)}{2p_i(1-p_i)^2(1-2p_i) + W_{i,0}(1-(2p_i)^m)(1-p_i) + (1+W_{i,0}(2p_i)^m)(1-2p_i)}. \quad (10)$$

Let τ_i be the probability that a STA in the priority i class transmits during a time slot e . We then have

$$\tau_i = b_{i,-1,0} + \sum_{j=0}^{m-1} b_{i,j,0} + b_{i,m,0} = \frac{1}{1-p_i} b_{i,-1,0}. \quad (10)$$

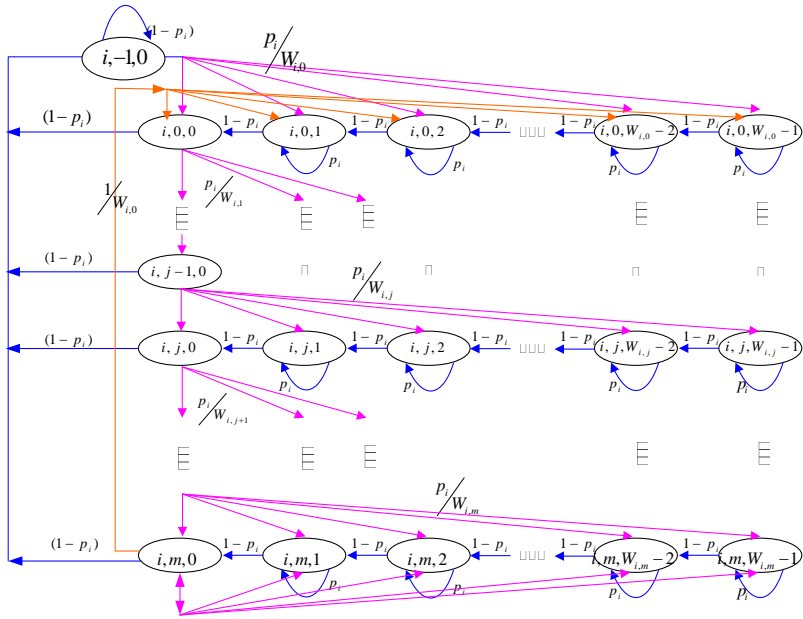


Fig. 1. Markov chain model for the priority i with the backoff window size

2.3 Block_Ack Mechanism

In IEEE 802.11e, a STA must receive an ACK frame to confirm a successful transmission. The Block_Ack is defined in order to reduce the channel wastes due to the ACK transmission. The Block_Ack mechanism improves channel efficiency by aggregating several ACKs into one frame. This mechanism allows a block of frames to be transmitted, each separated by a SIFS period, and to be acknowledged by a final aggregated ACK frame, which is called Block_Ack (B_ACK) frame as shown in Fig. 2. The Block ACK contains information about the reception of the whole block through a corresponding bitmap, and it is transmitted after an explicit transmitter makes a request, which is named Block ACK Request (BAR). In the B_ACK mechanism, the sender STA only contends for the channel access before the first frame of a block. If the STA wins the channel access, the STA sends out a whole block and a Block ACK Request frame and then waits for the Block ACK frame. Upon receiving the B_ACK frame correctly, the STA must defer a DIFS interval and a backoff process before sensing the channel again. At the same time, the other STAs should wait until the STA ends the B_ACK mechanism and then defer another DIFS interval before counting down their backoff counters for the next round of contention.

In general, the STA sends out a whole block of frames and a BAR frame. If the receiver STA detects a collision, the receiver STA will not send back the B_ACK frame. If the sender STA cannot receive the B_ACK frame, it must retry the transmission again.

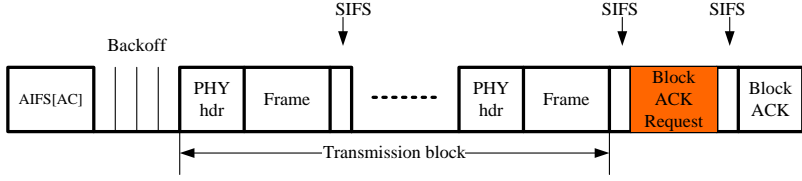


Fig. 2. The B_ACK frame structure

3 Description of the system performance

The saturation throughput is defined as the frame information in a slot time that is successfully transmitted in slot duration. There are four types of states in the B_ACK mechanism: 1) None of the STAs transmit at any time slot, i.e., the channel is idle. 2) Only one STA transmits at the time slot and the transmission is successful. 3) There is at least two STAs transmit at the same time slot, i.e., the channel has a collision. 4) There is at least one frame corrupted in a block, i.e., the channel has an error. We then can obtain that

$$S_i = \frac{p_{s,i}(1-P_{e,i})N_iT_L}{(1-p_i)\sigma + \sum_{i=0}^{N-1} p_{s,i}(1-P_{e,i})T_{S,i} + [p_i - p_{s,i}]T_{C,i} + \sum_{i=0}^{N-1} p_{s,i}(P_{e,i}^{RTS}T_{e,i}^{RTS} + P_{e,i}^{CTS}T_{e,i}^{CTS} + P_{e,i}^L T_{e,i}^L)} \quad (12)$$

$$T_{S,i} = T_{RTS} + T_{CTS} + 2T_{SIFS} + N_i(T_L + T_{SIFS}) + AIFS(i) + (T_{BAR} + T_{SIFS} + T_{BA}) + (N_i + 2)T_H \quad (13)$$

$$\begin{aligned} T_{C,i} &= T_{RTS} + T_{CTS} + 2T_{SIFS} + N_i(T_L + T_{SIFS}) + T_{BAR} + (N_i + 1)T_H + T_{EIFS,i}, \\ T_{EIFS,i} &= T_H + T_{BA} + AIFS(i) + T_{SIFS}. \end{aligned} \quad (14)$$

Where N_i denotes the number of frames in a block. $T_{S,i}$ is the time of the whole block transmission due to a successful transmission for priority i class. σ is the duration of an empty time slot.

4 Simulation Results

The following simulation results have been obtained assuming the parameters are as follows: Frame payload = 1023 bytes, MAC header = 34 bytes, PHY header = 16 bytes, ACK = 14 bytes, RTS = 20 bytes, CTS = 14 bytes, SIFS = 20 μ s, DIFS = 50 μ s, propagation delay = 1 μ s, slot time = 20 μ s, and $[W_{0,0} \ W_{1,0} \ W_{2,0} \ W_{3,0}] = [16 \ 32 \ 64 \ 128]$.

Fig. 3 and Fig. 4 show the saturation throughput performances of different priority STAs using the RTS/CTS CSMA/CA scheme with/without the B_ACK mechanism under a non-ideal channel scenario. From the results, we can conclude that the STA with B_ACK mechanism got better performance than that without B_ACK mechanism because the B_ACK mechanism improves channel efficiency by aggregating several ACKs into one frame.

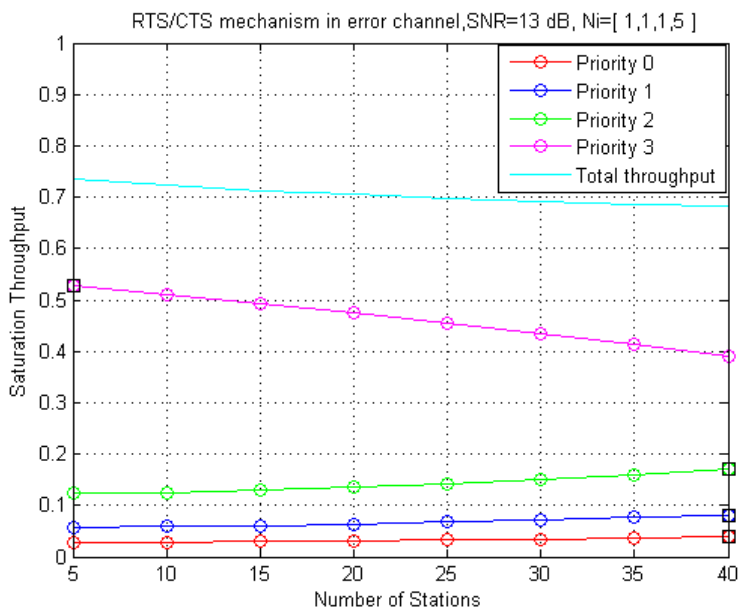


Fig. 3. Saturation throughput of the different priority STAs with the B_ACK mechanism

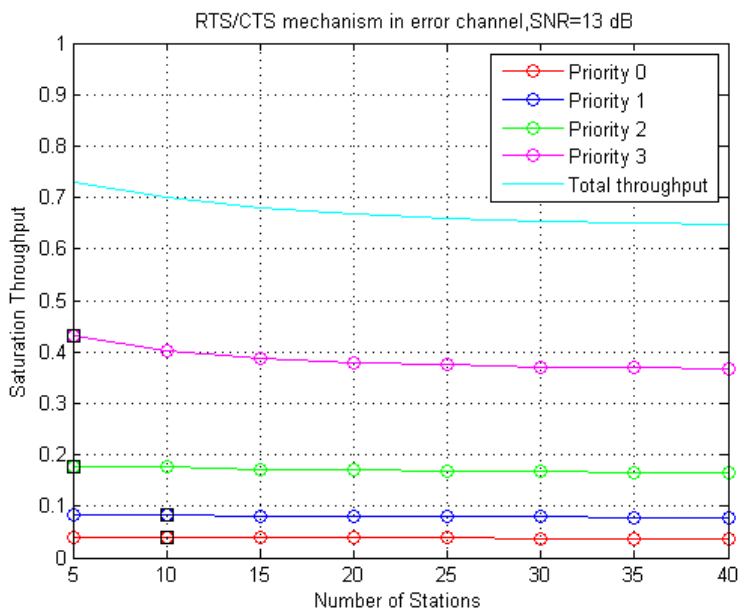


Fig. 4. Saturation throughput of the different priority STAs without the B_ACK mechanism

5 Conclusions

In this paper, we studied the backoff process characteristics of Markov chain model with RTS/CTS CSMA/CA scheme under non-ideal channel scenario. In order to increase the system throughput we integrated the B_ACK mechanism into the model. The model with the B_ACK mechanism under a channel error scenario can achieve a higher performance than that without the B_ACK mechanism due to fact that the model with the B_ACK mechanism is allowed to aggregate several ACKs into one frame.

Reference

1. Y Xiao, "QoS guarantee and provisioning at the contention-based wireless MAC layer in the IEEE 802.11e wireless LANs," *IEEE Wireless Communications Magazine*, vol. 13, pp. 14-21, Feb. 2006.
2. Daqing Gu and Jinyun Zhang, "QoS enhancement in IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 41, pp. 120-124, June 2003.
3. S. Choi, J. del Prado, S. Shankar N, and S. Mangold, "IEEE 802.11e contention-based channel access (EDCF) performance evaluation," *Proceedings of the IEEE International Conference on Communications (ICC 2003)*, May 2003.
4. D. J. Deng and R. S. Chang, "A priority Scheme for IEEE 802.11 DCF Access Method," *IEICE Trans. Communications*, Vol. E82-B, No.1, Jan. 1999, pp.96-102.
5. I. Aad and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," *IEEE INFOCOM 2001*.
6. A. Veres, A. T. Campbell, M. Barry, and L.H. Sun, "Supporting Differentiation in Wireless Packet Networks Using Distributed Control," *IEEE J-SAC*, Vol. 19. No. 10, pp. 2081-2093, Oct. 2001.
7. Xin XIA, "Enhanced DCF MAC scheme for providing differentiated QoS in ITS," *Proceedings of IEEE Intelligent Transportation Systems Conference*, pp. 280-286, Oct. 2004.
8. Daqing Gu and Jinyun Zhang, "QoS enhancement in IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 41, pp. 120-124, June 2003.
9. Jie Ma, Yuanan Liu, and Bihua Tang, "QoS research and design for WLAN," *Proceedings of IEEE International Symposium on Communications and Information Technology*, vol. 2, pp. 865-868, Oct. 2005.
10. Tetsu Nakajima, Tomoko Adachi, Yasuyuki Nishibayashi, Yoriko Utsunomiya, Tomoya Tandai, and Masahiro Takagi, "Compressed block ACK, an efficient selective repeat mechanism for IEEE 802.11n," *IEICE Tech. Rep.*, CS2004-194, pp. 65-70, Jan. 2005.